

Performance Testing Plan for Streamlining Ticket Support Operations

Date	2-11-2025
Team id	NM2025TMID04066
Project name	Streamlining ticket support for efficient support operation
Team members	4

Performance Testing Plan for Streamlining Ticket Support Operations

1. Objective

To verify that the optimized ticket support system:

- Handles **expected and peak ticket loads** efficiently
 - Maintains **stability, responsiveness, and scalability** under stress
 - Meets defined **SLAs and KPIs** (e.g., response time, uptime, throughput)
 - Ensures smooth **agent and customer experience** post-streamlining
-

2. Scope

In Scope:

- Ticket creation and routing workflows
- Automation & AI modules (auto-triage, response suggestion, tagging)
- Knowledge base search and retrieval
- Agent dashboard and customer self-service portal
- API integrations (CRM, chat, email)
- Analytics/reporting dashboards

Out of Scope (optional):

- External third-party systems not directly controlled by your org
 - Cosmetic UI performance issues unless they affect ticket resolution times
-

3. Key Performance Testing Types

Test Type	Purpose	Example Scenario
Load Testing	Validate system performance under expected daily load.	5,000 concurrent tickets created per hour.

Performance Testing Plan for Streamlining Ticket Support Operations

Test Type	Purpose	Example Scenario
Stress Testing	Determine system breaking point under extreme conditions.	Spike to 10x daily ticket volume.
Scalability Testing	Check performance as more servers or resources are added.	Add cloud instances to handle 50% more load.
Endurance (Soak) Testing	Assess performance over long durations.	Simulate 72-hour continuous ticket traffic.
Spike Testing	Observe system response to sudden workload surges.	Outage causes 2,000 tickets in 5 minutes.
Concurrency Testing	Validate simultaneous user activity handling.	200 agents updating tickets at once.
Integration Performance Testing	Ensure APIs and data sync remain performant.	Measure response time between ticket system and CRM.

4. Key Performance Metrics (KPIs)

Metric	Target Benchmark	Why It Matters
Ticket Creation Time	< 2 seconds	Affects customer experience.
Ticket Assignment (Routing) Time	< 1 second	Ensures real-time auto-triage efficiency.
Dashboard Load Time	< 3 seconds	Affects agent productivity.
Average Response Time (System API)	< 200 ms	Ensures quick data retrieval.
Throughput	1,000 tickets/minute (baseline)	Determines max processing capacity.
Error Rate	< 0.5%	Ensures stability.
System Uptime	≥ 99.9%	Supports SLA compliance.
Memory & CPU Utilization	< 70% under average load	Ensures resource efficiency.

5. Test Environment Setup

Performance Testing Plan for Streamlining Ticket Support Operations

Component	Details
Test Environment	Replica of production environment (servers, databases, load balancers).
Test Data	Synthetic and anonymized ticket data across multiple categories.
User Simulation	Virtual agents/customers mimicking real usage patterns.
Monitoring Tools	Application monitoring (e.g., New Relic, Dynatrace, Grafana).
Load Generation Tools	JMeter, LoadRunner, Locust, or k6.

6. Sample Test Scenarios

Scenario ID	Scenario Description	Expected Result
PT-001	1,000 tickets created per minute for 60 minutes	System remains stable, no lag or crash
PT-002	200 agents update tickets simultaneously	CPU < 70%, no data loss or timeout
PT-003	2,000 ticket surge in 5 minutes	Auto-triage and routing complete within 1 second/ticket
PT-004	Continuous 72-hour load at 60% of daily max	No memory leaks, stable throughput
PT-005	Real-time analytics dashboard refresh under high load	Load time < 3 seconds

7. Tools and Technologies

Category	Recommended Tools
Load Testing	Apache JMeter, LoadRunner, Locust, k6
Monitoring & Logging	Grafana, Prometheus, ELK Stack
Cloud Performance Testing	AWS CloudWatch, Azure Load Testing, Google Cloud Operations Suite
API Testing	Postman, SoapUI
Issue Tracking	Jira, TestRail for test case management

Performance Testing Plan for Streamlining Ticket Support Operations

This document outlines the performance testing plan for streamlining ticket support operations. The goal is to identify bottlenecks and areas for improvement in the current ticket handling process.

Test Objectives

- Identify performance bottlenecks in the ticket creation, assignment, and resolution processes.
- Assess the impact of ticket volume on system response times.
- Evaluate the efficiency of current ticket routing logic.
- Test the scalability of the ticket database under high load.
- Measure the time taken for ticket status updates and notifications.

Test Environment

The tests will be conducted on a simulated environment with the following characteristics:

- System Configuration:** Production hardware and software stack.
- User Load:** Simulated user activity with varying ticket volumes (low, medium, high).
- Network Conditions:** Standard office network conditions.
- Database:** Production database with historical ticket data.

Test Cases

The following test cases will be executed to validate the system's performance:

- Ticket Creation:** Create a large volume of tickets simultaneously to test the system's ability to handle concurrent requests.
- Assignment:** Assign tickets to agents to test the distribution logic and its impact on response times.
- Resolution:** Resolve tickets to test the system's performance in processing and updating ticket status.
- Reporting:** Generate reports to test the system's performance in generating large amounts of data.

Performance Metrics

The following metrics will be tracked during the performance testing:

- Response Time: Time taken to process and return a ticket.
- Throughput: Number of tickets processed per unit of time.
- Error Rate: Percentage of failed or incomplete ticket processing.
- Resource Utilization: CPU, memory, and disk usage.
- Latency: Time taken for ticket status updates and notifications.

Conclusion

The performance testing plan will help us identify areas for optimization and ensure that the ticket support operations remain efficient and reliable under various load conditions.